

**Amendments to the Specification:**

**The title is amended as follows:**

ARCHITECTURE FOR CUSTOMIZABLE APPLICATIONS SUPPORTING EVENT  
HANDLING AND QUERYING CUSTOMIZABLE OBJECTS IN A DATABASE

**The paragraphs appearing on page 8, line 25 through page 10, line 3 are amended as follows:**

21  
The computer 110 may also include other removable/non-removable, volatile/nonvolatile computer storage media. By way of example only, FIG. 1 illustrates a hard disk drive 140 that reads from or writes to non-removable, nonvolatile magnetic media, a magnetic disk drive 151 that reads from or writes to a removable, nonvolatile magnetic disk 152, and an optical disk drive 155 that reads from or writes to a removable, nonvolatile optical disk 156, such as a CD ROM or other optical media. Other removable/non-removable, volatile/nonvolatile computer storage media that can be used in the exemplary operating environment include, but are not limited to, magnetic tape cassettes, flash memory cards, digital versatile disks, digital video tape, solid state RAM, solid state ROM, and the like. The hard disk drive 141 is typically connected to the system bus 121 through an non-removable memory interface such as interface 140, and magnetic disk drive 151 and optical disk drive 155 are typically connected to the system ~~bus 121~~ by bus 121 by a removable memory interface, such as interface 150.

Am The drives and their associated computer storage media discussed above and illustrated in FIG. 1, provide storage of computer readable instructions, data structures, program modules and other data for the computer 110. In FIG. 1, for example, hard disk drive 141 is illustrated as storing operating system 144, application programs 145, other program modules 146, and program data 147. Note that these components can either be the same as or different from operating system 134, application programs 135, other program modules 136, and program data 137. Operating system 144, application programs 145, other program modules 146, and program data 147 are given different numbers here to illustrate that, at a minimum, they are different copies. A user may enter commands and information into the computer ~~20~~ 110 through input devices such as a keyboard 162 and pointing device 161, commonly referred to as a mouse, trackball or touch pad. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit 120 through a user input interface 160 that is coupled to the system bus, but may be connected by other interface and bus structures, such as a parallel port, game port or a universal serial bus (USB). A monitor 191 or other type of display device is also connected to the system bus 121 via an interface, such as a video interface 190. In addition to the monitor, computers may also include other peripheral output devices such as speakers 197 and printer 196, which may be connected through an output peripheral interface ~~190~~ 195.

---

The paragraph appearing at page 15, lines 12-18 is amended as follows:

A2  
FIG. 4 depicts the structure of application object 201a as a collection of instructions including both predetermined actions 401 and events 312. Application object 201a includes executable code that executes in a sequence ~~401(1)~~ 410(1). FIG. 4 shows a plurality of code sequences ~~401(1)~~ 410(1) through ~~401(n)~~ 410(n), because application object 201a may, in fact, have plural threads of execution that execute concurrently, and thus not all of the code of application object 201a can be viewed as executing in a single sequence.

The paragraph appearing at page 22, line 18 through page 23, line 3 is amended as follows:

A3  
The process of creating a moniker 800 from which a customization 320 can be located and retrieved is now described with reference to FIG. 10. After application object 201a starts, it gathers or refers to its own fixed data ~~801~~ 802 (step 1001). For example, application object 201a may collect the data describing the name of the company in which it is installed, the names of the application and module of which it is a part, and its own name. At step 1002, application object 201a proceeds to gather environmental data ~~802~~ 803 by querying operating environment 901. This may be done, for example, by operating system routines that return the attributes of the currently-logged-in user, such as his login-name and the organization of which he is a part. At step 1003, the moniker string 800 is constructed using the fixed data ~~801~~ 802 and the environmentally-obtained data ~~802~~ 803. As discussed above, FIG. 8 shows an exemplary format of the moniker string 800, but a variety of formats could be used as well. It will be observed that

**DOCKET NO.:** MSFT-0231/160306.1

**PATENT**

**Application No.:** 09/678,511

**Office Action Dated:** April 25, 2003

*83*  
the moniker string 800 shown in FIG. 8 is organized hierarchically, which may be a convenient  
format for monikers.

---